Senior Design
ENG EC 463

Test Report - Prototype 2.0

Team 21 - IoT Kitchen

Team Members
Addison Dolido addison@bu.edu
Erin Dorsey edorsey1@bu.edu
Saransh Kothari saranshk@bu.edu
Yuran Shi yuran@bu.edu
Kenny Zheng ykzheng2@bu.edu

Submitted: February 29th, 2020

Application Navigation

Equipment & Setup

The equipment required for this test consists of an Android mobile device with IoT_Kitchen application installed. Once installed, users should have Bluetooth turned on for compatibility to receive data from the temperature sensor and scale. Users will be prompted to then sign in and navigate throughout the application.

Results

The measurable criteria are as follow:

1. Mobile Application does not crash **(Yes/No)**
2. Each navigation bar shows a new screen **(Yes/No)**
3. Recipes shows recipe you wish to make **(Yes/No)**
4. RECIPES directs you over to COOK **(Yes/No)**

Test 1 should be successful if the mobile application does not crash upon starting. The application should be able to run successfully with the implementation of the navigation bar. Test 2 should be successful if the navigation bar includes the following: Device Screen, Cook, and Recipes. This follows test 3 and 4 which should show you the recipe of the recipe you wish to make which should then direct you over to the cooking page. For example, if you select recipes and select Peanut Butter Cups; prompting you over to Cook page that includes the following: instructions, procedure, unit and weight.

Conclusions

The navigation bar was successfully implemented onto IoT_Kitchen offering users the option to easily navigate through their cooking process. Each test was carefully tested with great success rate.

Natural Language Understanding Change Recipe

Equipment & Setup

The equipment required for this test are a computer with internet connectivity, an Android device loaded with the IoT Kitchen Android application, the iotk-nlu-test Dialogflow console, the Google Assistant emulator, and the IotKitchen-NLU-Test Google Firebase Cloud Firestore database console. To set up the test, on the computer team 21 loaded the project on the Dialogflow console at https://dialogflow.cloud.google.com and the project database pm https://console.firebase.google.com/. Then the Google Assistant emulator is loaded from the Dialogflow console by selecting “Google Assistant”. The Dialogflow powered agent was launched by pressing the microphone button and verbally inputting “Talk to my test app”. Test

phrases were also entered by pressing the microphone button and providing verbal input. After interacting with the Dialogflow agent, the test was checked by loading the IoT Kitchen Android app and navigating to the tested recipe page.

Results

The measurable criteria was as follows:

1. Module has response for all phrases **(Yes/No)**
2. Intent changes the correct database field **(Yes/No)**
3. Intent changes field to match user input **(Yes/No)**
4. Database changes are reflected on application after 1 refresh **(Yes/No)**
5. Module does not crash **(Yes/No)**

For test 1, the test would be considered a success if all phrases were matched to a response that was not the default fallback response ("I didn't understand." or "Could you repeat that please?") given if the agent did not understand or have a response to input . Test 2 would be considered a success if each test phrase matched the intent that changed the intended field in the database. That is, if the user specifies "Change instruction 1" and "Ingredient is...", then the ingredient field is changed in step1. Test 3 is a success if in that specified database field, the value of the key value pair is changed to match the user input. For example, if the user says "Procedure is mix ingredients together", the procedure field will be given the value of "mix ingredients together". Test 4 would be considered a success if after all test phrases are completed, the recipe page of IoT Kitchen Android application will show all changes made by the user after refreshing the application one time. Finally, test 5 is considered a success if the module did not crash at any point. A crash is indicated by the Dialogflow agent returning a response of "Not available".

Before executing the test, the data base value of PeanutButterCups.instruction.step2 and PeanutButterCups.ingredient.i2 were recorded as follows:

```
i2: 130 g of creamy peanut butter
step2:
    ingredient: creamy peanut butter
    number: 2
    procedure: add peanut butter to bowl
    unit: g
    weight: 130
```

After the test was executed, the database read as follows:

```
i2: 120 g of crunchy peanut butter
step2:
    ingredient: crunchy peanut butter
    number: 2
```

procedure: melt peanut butter and add to bowl
unit: g
weight: 120

After the test, the recipe page for Peanut Butter Cups had the following values for ingredient 2 and instruction 2:

I2: 120 g of crunchy peanut butter
Step2: melt peanut butter and add to bowl

Conclusions

The Dialogflow agent was able to succeed in all tests and achieve the test criteria. The module was able to provide the correct response to all phrases, did not crash at any point, and updated the database with the correct values of user input. This test shows that this module will allow users to update their recipes easily and have their changes reflected on both the app and the database in real time.

1.0 Temperature Sensor - Application Integration/Accuracy

Equipment & Setup

The equipment required for this test are a computer, an Android device with bluetooth enabled and the IoT Kitchen Android application installed, a medical under the tongue thermometer, and the temperature sensor prototype, which consists of a k-type thermocouple, a MAX 31856 amplifier, Arduino Micro, LCD display, a HC-05 bluetooth module, and a USB to micro USB cable. The test begins by plugging in the temperature sensor prototype into the computer and recording its initial temperature reading of the room displayed on the LCD. Then the medical thermometer is turned on and put under the participants tongue with the k-type thermocouple from the prototype. Then observe the readings from both devices and record the time it takes for the temperature sensor to stabilize its reading and the final temperatures displayed by both devices once the medical thermometer beeps to indicate it is finished. Finally, remove the temperature sensor and record the time it takes for it to return to its initial room temperature reading.

Results

The measurable criteria was as follows:

1. LCD displays the temperature **(Yes/No)**
2. App displays and updates the same temperature as that on the LCD **(Yes/No)**
    a. Percent difference between the temperature read by the medical thermometer and the prototype

    b. Time it takes for the temperature sensor to stabilize its reading from being placed under the participant's tongue
3. Does the temperature sensor return to near its initial room temperature reading after being removed from the participant's mouth **(Yes/No)**
    a. Percent difference between the initial room temperature reading and the new reading
    b. Time it takes for the temperature sensor to return to room temperature
4. Can the temperature be read in phone application **(Yes/No)**

We ran this test 3 times, once presenting and twice in preparation all using the same participant for sanitation reasons. The results are below:

1. All 3 tests had the LCD display temperature properly.
2.

Test number	Medical thermometer reading (F)	Temperature Sensor reading (C)	Temperature Sensor reading converted (F)	Percent difference (%)	Time to reach stable reading (s)
1	96.8	36	96.8	0	10
2	96.7	36	96.8	0.1	12
3	96.8	36	96.8	0	9

3.

Test number	Initial room temperature reading (C)	Final room temperature reading (C)	Percent difference (%)	Time to reach stable reading (s)
1	21	21	0	8
2	21	21	0	13
3	21	21	0	11

4. The temperature was displayed in the application during all 3 tests

Conclusions

The temperature sensor succeeded in all of the tests proving that it is accurate at temperatures ranging from 20 - 100 C to the nearest degree. The tests also proved the water resistance of the sensor by being put into the participants mouth and being able to read the

proper temperature during and after. Plus, the temperature sensor reached a stable point much faster than the medical thermometer, which took between 30 seconds to 1 minute to indicate that it was finished.